NASA/TM—2002-211710

An Interactive Excel Program for Tracking a Single Droplet in Crossflow Computation

E. Urip and S.L. Yang
Michigan Technological University, Houghton, Michigan

C.J. Marek
Glenn Research Center, Cleveland, Ohio

August 2002

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at 301–621–0134

- Telephone the NASA Access Help Desk at 301–621–0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076

NASA/TM—2002-211710

An Interactive Excel Program for Tracking a Single Droplet in Crossflow Computation

E. Urip and S.L. Yang
Michigan Technological University, Houghton, Michigan

C.J. Marek
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center

August 2002

Acknowledgments

Undertaking this project has given me (Egel Urip) an enormous knowledge in spray dynamics, and it has given me a chance to employ my skills in solving problems of spray in a Crossflow. I am very grateful for being given this project. I would like take this opportunity to thank my graduate advisor, Dr. S.L. (Jason) Yang, for all his support in solving differential equations; Dr. Cecil J. Marek for his helpful and useful comments. I was truly honored to have worked with these experienced and friendly people. I hope this code will help engineers in exploring the spray in crossflow application. I also hope that it will provide an example for using Excel for other problems.
The authors would like to thank Dr. Paul Penko for being the grant monitor on this project; and Dr. William F. Ford of the NASA Glenn Computer Services Division for helping with many details of the code such as using emergency halts, simplifying the code, and controlling the sheet size.

The Aerospace Propulsion and Power Program at NASA Glenn Research Center sponsored this work.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at http://gltrs.grc.nasa.gov/GLTRS

An Interactive Excel Program for Tracking a Single Droplet in Crossflow Computation

E. Urip and S.L. Yang
Michigan Technological University
Houghton, Michigan 49931
Ph.: 906–487–2624
Fax: 906–487–2822
E-mail: slyang@mtu.edu

C.J. Marek
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135
Ph.: 216–433–3584
Fax: 216–433–3000
E-mail: Cecil.J.Marek@grc.nasa.gov

Abstract

Spray jet in crossflow has been a subject of research because of its wide application in systems involving pollutant dispersion, jet mixing in the dilution zone of combustors, and fuel injection strategies. The focus of this work is to investigate dispersion of a 2-dimensional atomized spray jet into a 2-dimensional crossflow. A quick computational method is developed using available software. The spreadsheet can be used for any 2D droplet trajectory problem where the drop is injected into the free stream eventually coming to the free stream conditions.

During the transverse injection of a spray into high velocity airflow, the droplets (carried along and deflected by a gaseous stream of co-flowing air) are subjected to forces that affect their motion in the flow field. Based on the Newton’s Second Law of motion, four ordinary differential equations were used. These equations were then solved by a 4^{th}-order Runge-Kutta method using Excel software.

Visual basic programming and Excel macrocode to produce the data facilitate Excel software to plot graphs describing the droplet’s motion in the flow field. This program computes and plots the data sequentially without forcing users to open other types of plotting programs. A user’s manual on how to use the program is also included in this report.

Nomenclature

A_d Projected area of the droplet, πr^2

C_D Drag coefficient, *see equation (8)*

$\vec{F}$ Force

g Gravitational acceleration constant, $9.8 m/s^2$

n n^{th} iteration

Re Reynolds number, *see equation (9)*

r Spherical radius of the droplet

u_g Velocity of the cross stream (air) in the x-direction

u_d Velocity of the droplet in the x-direction

$\vec{U}$ Velocity

$\vec{U}_R$ Relative velocity between the droplet and the gas stream

$$\vec{U}_R = (u_d - u_g)\hat{i} + (w_d - w_g)\hat{k}$$

V_d Droplet Volume, $\frac{4}{3}\pi r^3$

w_g Velocity of the cross stream (air) in the z-direction

w_d Velocity of the droplet in the z-direction

x horizontal direction ± equals to the right

z vertical direction ± equals up

ρ_g Density of the cross stream (air)

ρ_d Density of the droplet

μ_g Viscosity constant of the gaseous fluid

Δt time step-size

SUBSCRIPTS:

d Droplet

g Gaseous phase

Introduction and Governing Equations

A liquid spray injected into a gaseous crossflow is important because of its wide application in systems involving two phase mixing and in combustion requiring quick mixing and reduction of pollutants, for jet mixing in the dilution zone of combustors, and for determining fuel injection strategies. It is important to be able to compute this flow to optimize the mixing strategy.

This work is mainly focused on producing a quick computational method for determining spray penetration. With this spreadsheet, one can investigate the dispersion of an air-blast atomized spray jet into a crossflow, see Figure 1. During the transverse injection of a spray into high velocity airflow, the droplets (carried along in the gaseous stream of co-flowing air) are subjected to forces that affect their motion in the flow field (see Figure 2).

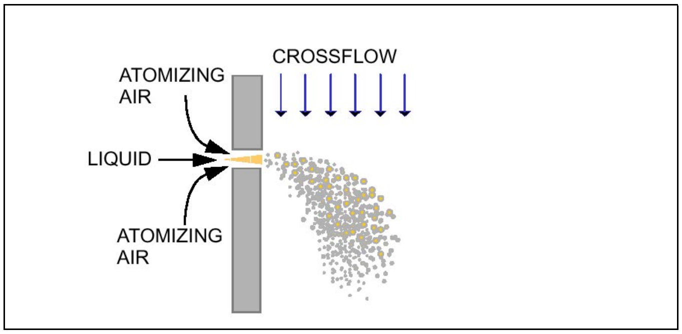

Figure 1 (from NASA/CR—2000-210467, reference 1)

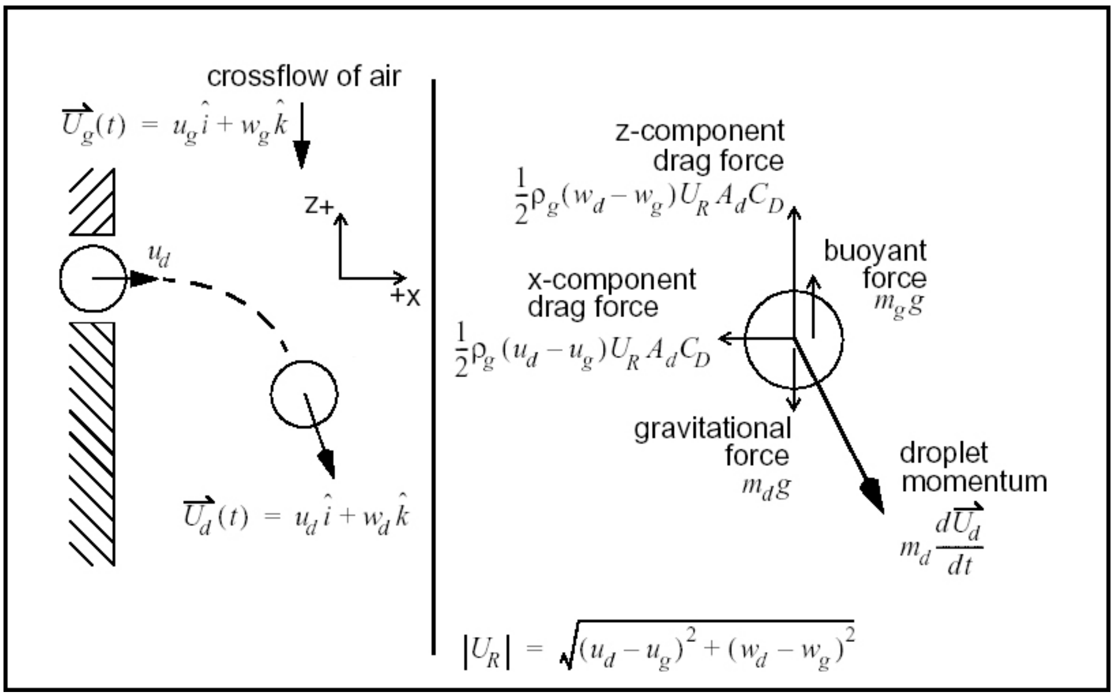

Figure 2 (from NASA/CR—2000-210467)

The trajectories of the droplets can be tracked by applying a Lagrangian-based analysis to the droplets. Since no evaporation is assumed, the code can be used for solid particles as well. The momentum equations for a droplet can be obtained by equating the droplet motion to: (1) The viscosity and pressure-related drag forces, (2) The pressure gradient and viscous forces related to the fluid surrounding the droplet, (3) The inertia of the virtual mass, induced when the particle acceleration affects the fluid mass acceleration, and (4) The Basset force, which takes into account the acceleration history of the droplet.

Based on these principles along with the following assumptions:

(1) The droplets are spherical,
(2) No droplets breakup occurs,

(3) Vaporization is not considered and is assumed negligible, and
(4) Lift, virtual mass, and Basset forces are neglected.
(5) Chemical reaction is not included,

droplet trajectory and velocity with respect to time can be calculated. These assumptions reduce the droplet momentum equation to include only the effects of the drag and body forces. The general momentum equations for a single droplet injected along the positive x-direction, transversely into a downward-flowing air stream in the positive z-direction, as shown in Figure 2, is described by

$$\vec{F}_d = \vec{F}_{drag} + \vec{F}_{body} \tag{1}$$

where the net force $\vec{F}_d$ that drives the droplet motion is balanced by the drag force opposing its motion, and the field forces acting on the droplet. The aerodynamic drag force is given by

$$\vec{F}_{drag} = -\frac{1}{2}\rho_g \vec{U}_R \left|\vec{U}_R\right| A_d C_D \tag{2}$$

where ρ_g is the air density, and A_d and C_D, the projected area and the drag coefficient of the droplet, respectively. The relative velocity between the droplet and the crossflow has a magnitude of U_R (see Figure 2). The subscript "*d*" refers to the droplet and "*g*" the crossflow air.

The body force, resulting from an equivalent volume of air that buoys the droplet, includes the gravitational and buoyancy forces. It is given by

$$\vec{F}_{body} = \left(\rho_g - \rho_d\right) V_d \ \vec{g} \tag{3}$$

which says that the body force is equal to the product of relative droplet and air density (ρ_d - ρ_g), the droplet volume V_d, and the gravitational acceleration g.

Substituting equation 2 and equation 3 to equation 1 yields:

$$\rho_d V_d \frac{du_d}{dt} = -\frac{1}{2}\rho_g \left(u_d - u_g\right) \left|\vec{U}_R\right| A_d C_D \tag{4}$$

$$\rho_d V_d \frac{dw_d}{dt} = -\frac{1}{2}\rho_g \left(w_d - w_g\right) \left|\vec{U}_R\right| A_d C_D + \left(\rho_g - \rho_d\right) V_d \ g \tag{5}$$

$$\frac{dx}{dt} = u_d \tag{6}$$

$$\frac{dz}{dt} = w_d \tag{7}$$

The drag coefficient of the droplet depends on the droplet Reynolds number and is given by

$$C_D = \begin{cases} \dfrac{24}{\mathrm{Re}_d}\left[1 + \dfrac{1}{6}\mathrm{Re}_d^{\,2/3}\right] & \mathrm{Re}_d \le 1000 \\ \\ 0.424 & \mathrm{Re}_d > 1000 \end{cases} \tag{8}$$

where Re_d is the droplet Reynolds number and is defined as follows

$$\mathrm{Re}_d = \frac{2\rho_g \left|\vec{U}_R\right| r_d}{\mu_g} \tag{9}$$

in which r_d is the droplet radius and μ_g is the gas (air) viscosity.

Numerical Method

Four ordinary differential equations are to be solved, namely, equations 4, 5, 6 and 7, for the four dependent variables u_d, w_d, x and z. The droplet trajectory is defined by the set of x and z values. A 4th-Order Runge-Kutta explicit method[1] was used to solve these equations. The Runge-Kutta explicit method is an ideal numerical scheme for solving ordinary differential equations using Excel software. It is a self-starting method with good stability characteristic. The step-size can be changed as desired without any complications for higher-order schemes. For a set of two coupled equations, such as,

$$\frac{dx}{dt} = f(x,z,t) \tag{10a}$$

$$\frac{dz}{dt} = g(x,z,t) \tag{10b}$$

the 4th-order Runge-Kutta method reads (subscript n stands for the n^{th} time step); k and l are unknown constants.

[1] Nakamura, Shoichiro, *Applied Numerical Method With Software*, Englewood Cliffs, New Jersey: Prentice Hal, 1991.

$$x_{n+1} = x_n + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \tag{11a}$$

$$z_{n+1} = z_n + \frac{1}{6}\left(l_1 + 2l_2 + 2l_3 + l_4\right) \tag{11b}$$

where

$$\begin{aligned}
k_1 &= \Delta t \cdot f(x_n, z_n, t_n) \\
k_2 &= \Delta t \cdot f(x_n + \frac{k_1}{2}, z_n + \frac{l_1}{2}, t_n + \frac{\Delta t}{2}) \\
k_3 &= \Delta t \cdot f(x_n + \frac{k_2}{2}, z_n + \frac{l_2}{2}, t_n + \frac{\Delta t}{2}) \\
k_4 &= \Delta t \cdot f(x_n + k_3, z_n + l_3, t_n + \Delta t)
\end{aligned} \tag{12a}$$

and

$$\begin{aligned}
l_1 &= \Delta t \cdot g(x_n, z_n, t_n) \\
l_2 &= \Delta t \cdot g(x_n + \frac{k_1}{2}, z_n + \frac{l_1}{2}, t_n + \frac{\Delta t}{2}) \\
l_3 &= \Delta t \cdot g(x_n + \frac{k_2}{2}, z_n + \frac{l_2}{2}, t_n + \frac{\Delta t}{2}) \\
l_4 &= \Delta t \cdot g(x_n + k_3, z_n + l_3, t_n + \Delta t)
\end{aligned} \tag{12b}$$

Only every nth cycle (as specified by the user) is saved for plotting. This greatly saves on storage and increases the speed of post processing. We have chosen to enter the data in SI units in the unlocked cells. The required conversions are done in the locked cells. When the user becomes familiar with the spreadsheet, the spreadsheet can be unlocked, with password NASA, and the user can adapt the spreadsheet as required.

The predictions were compared with the data of reference 1 with good results, see Figures 3 and 4. The experimental data from Figure 9.4 in ref. 1 for cross flow jets without an air blast assist was used. Jet to crossflow momentum flux ratio is used in this study to determine the depth of the droplet penetration. Momentum flux ratio for single phase jet is given by

$$q_1 = \frac{\rho U^2\big|_{jet}}{\rho U^2\big|_{crossflow}},$$

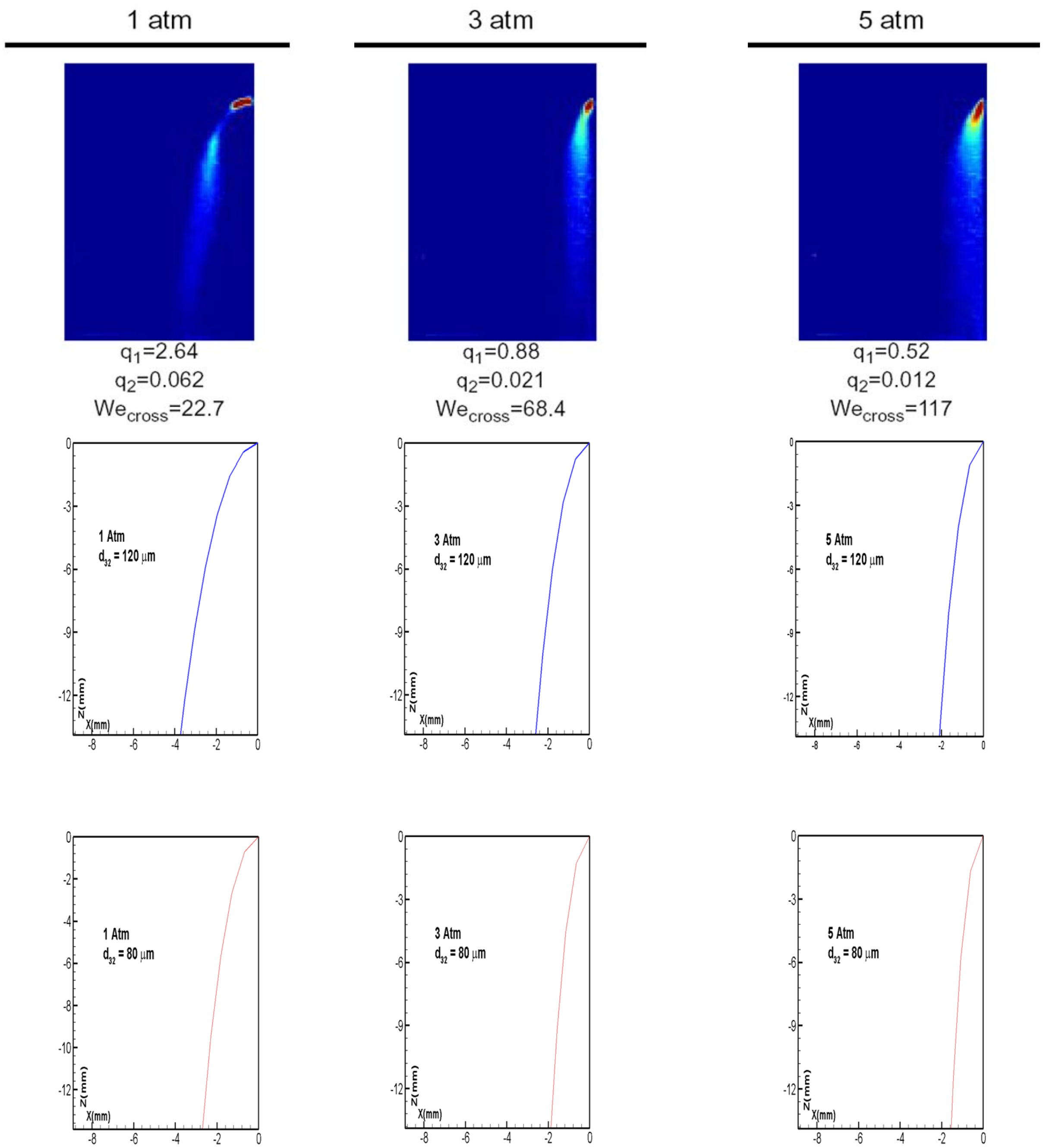

Figure 3—Experimental Comparison

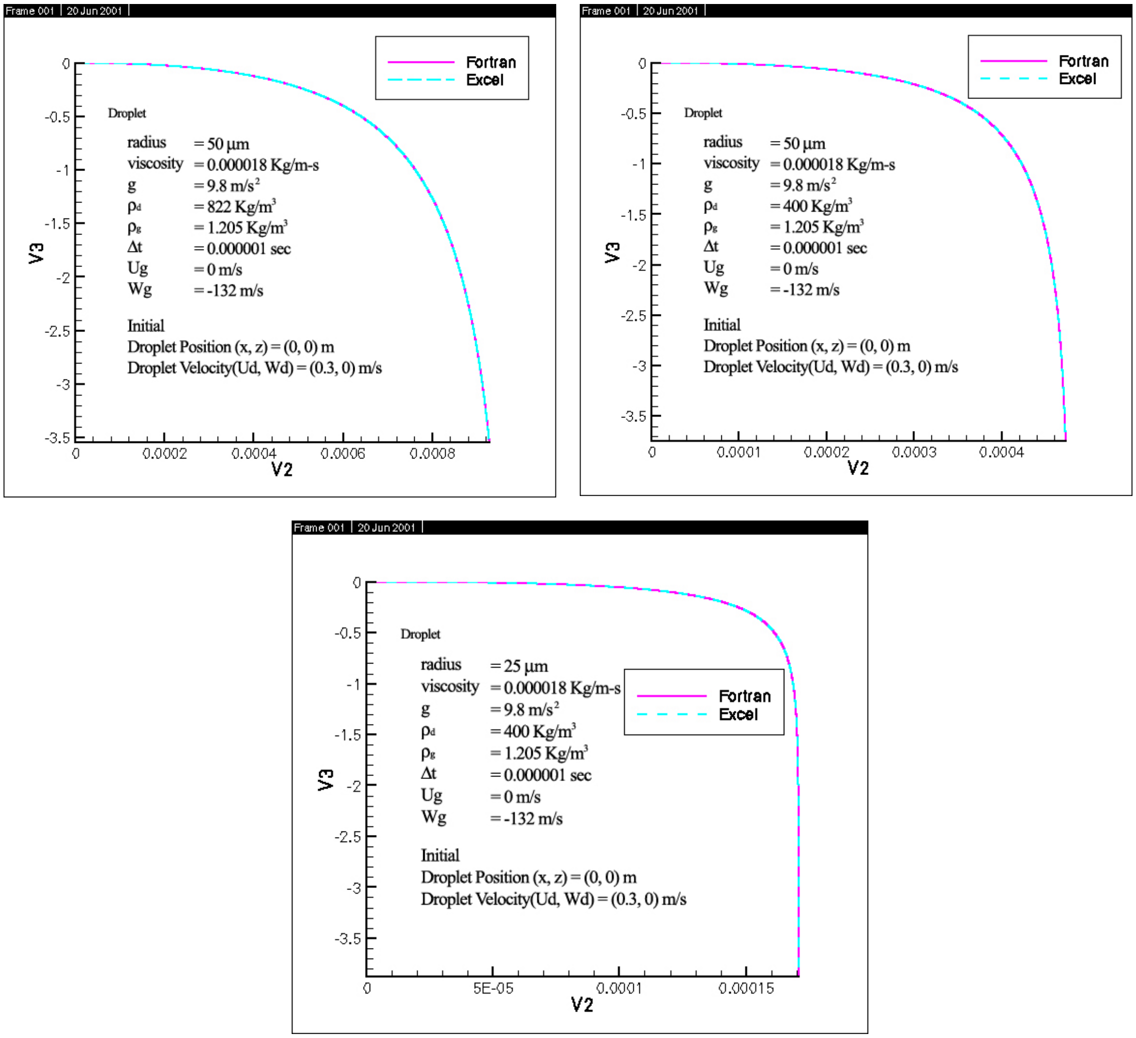

Figure 4—Droplet Trajectory Data Validation

and momentum flux ratio for two phase jet is given by

$$q_2 = \frac{\left(\rho_L U_L^2 A_{fuel} + \rho_g U_{airbl}^2 A_{airbl}\right) / A_{spray}}{\rho_g U_{cross}^2}$$

The 120 micron droplet predicted the most penetration and the 80 micron droplet predicted the least penetration.

To validate the results obtained using the Excel spreadsheet (because no analytical solution is available), a FORTRAN program was also developed for this purpose and is given in the appendix along with three validation cases (plots). FORTRAN and Excel calculations are compared in Figure 5. With an Excel spreadsheet, we do not have to compile, build, and link as in a regular FORTRAN code. In addition, the graphics are immediately displayed after the computations are completed, so that the results are seen quickly and changes in the input can be made. The interactive spreadsheet is available on this CD as a separate document. Additional copies of the spreadsheet can be requested by e-mailing: Cecil.J.Marek@grc.nasa.gov. The report portion can be accessed on the web at: http://gltrs.grc.nasa.gov/cgi-bin/GLTRS/browse.pl?2002/TM-2002-211710.html

User's Manual

This program is written in Microsoft Visual Basic Excel. There are three sheets in the program, namely the Instruction Sheet, the Process Sheet, and the Code Sheet.

Instructions Sheet

The instruction sheet contains a brief description of the problem, the solution method, and the user-input variables. Several schematics, which describe the forces applied on the droplet, can also be found on this sheet.

Process Sheet

The process sheet contains the user-inputs and the solution plots. There are five graphs on this sheet (see Figure 6), namely droplet trajectory, droplet velocity profile, drag coefficient, C_D, as a function of time, droplet velocity profiles as a function of time, and droplet trajectory profiles as a function of time. The cells highlighted in green are the user inputs. The ***cells highlighted in cyan*** contain computed values associated with the green cells; therefore, they are locked to prevent the user from modifying the values. When values are entered into the formula cells, the formulas are erased and linkages to other cells are interrupted. That is why for this version we chose to lock the closed cells without a password.

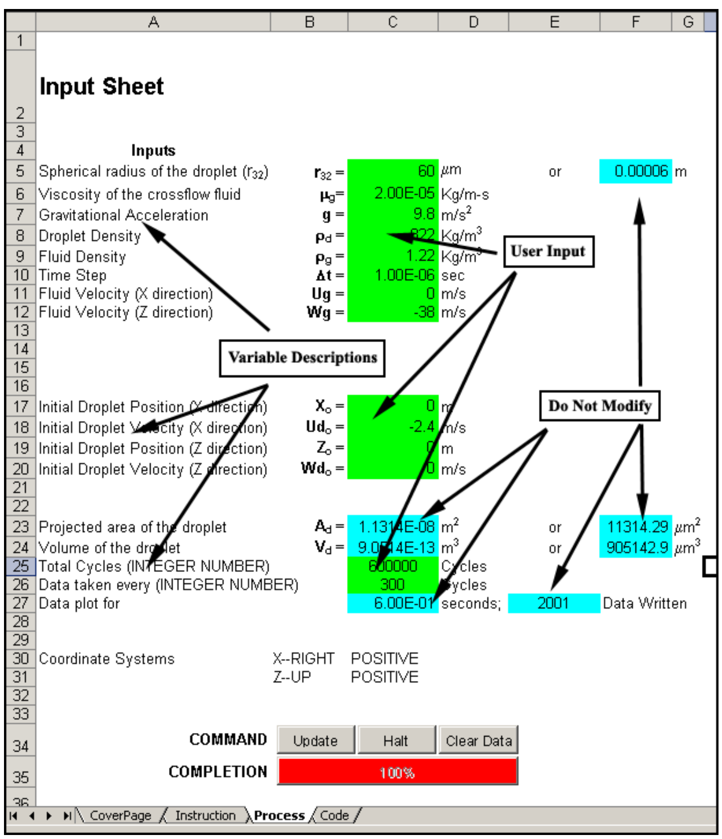

Figure 5

Input Sheet

Inputs

Spherical radius of the droplet (r_{32})	r_{32} =	60	m	or	0.00006	m
Viscosity of the crossflow fluid	$_g$=	2.00E-05	Kg/m-s			
Gravitational Acceleration	g =	9.8	m/s^2			
Droplet Density	$_d$=	822	Kg/m^3			
Fluid Density	$_g$=	1.22	Kg/m^3			
Time Step	t =	1.00E-06	sec			
Fluid Velocity (X direction)	Ug =	0	m/s			
Fluid Velocity (Z direction)	Wg =	-38	m/s			
Initial Droplet Position (X direction)	X_o =	0	m			
Initial Droplet Velocity (X direction)	Ud_o =	-2.4	m/s			
Initial Droplet Position (Z direction)	Z_o =	0	m			
Initial Droplet Velocity (Z direction)	Wd_o =	0	m/s			
Projected area of the droplet	A_d =	1.13143E-08	m^2	or	11314.29	m^2
Volume of the droplet	V_d =	9.05143E-13	m^3	or	905142.9	m^3
Total Cycles (INTEGER NUMBER)		600000	Cycles			
Data taken every (INTEGER NUMBER)		300	Cycles			
Data plot for		6.00E-01	seconds;	2001	Data Written	

Coordinate Systems X--RIGHT POSITIVE
Z--UP POSITIVE

COMMAND Update Halt Clear

COMPLETION 100%

OPTIONAL: For TecPlot Viewer ONLY
Do you want to save the data into a file? Yes
No

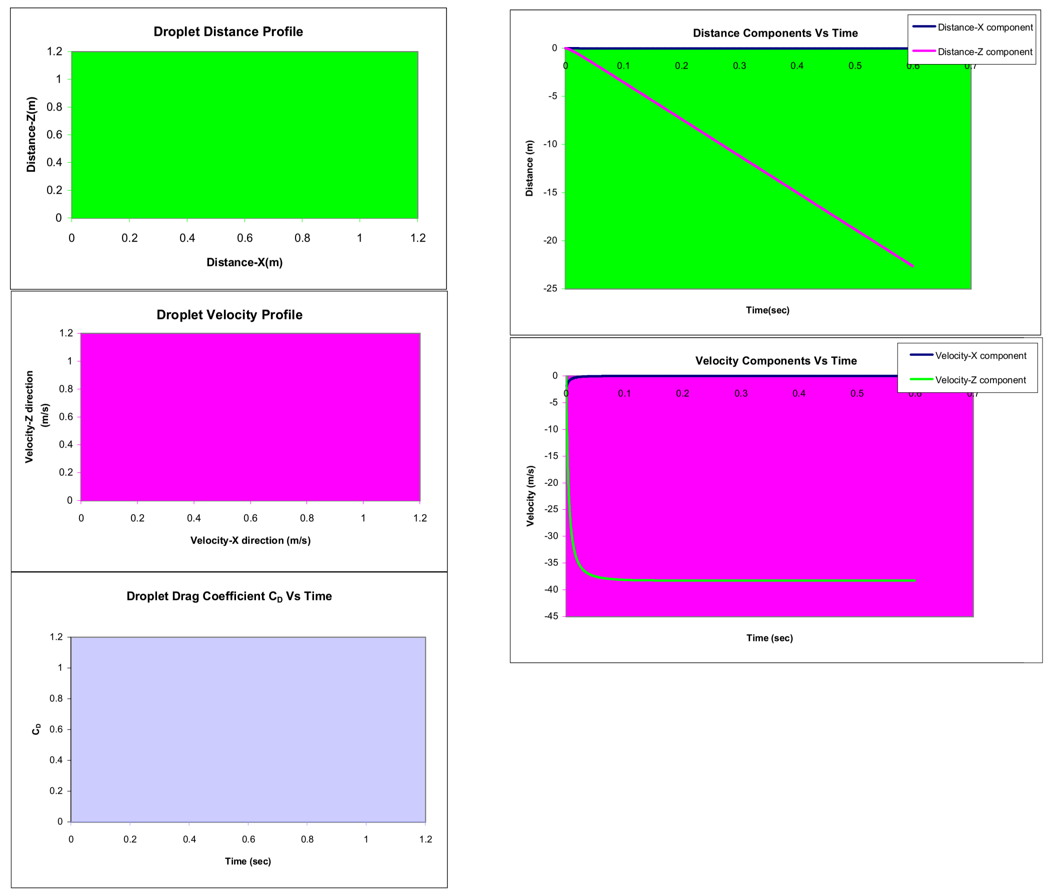

Figure 6

From Figure 5, the last two user inputs, ***Total Cycles (C25)*** and ***Data taken every ### cycles (C26)*** are included. ***The number assigned in cyan cell "E27"must be kept below 65,536; the cell will turn into red if this condition is not satisfied.*** Cell E27 basically shows amount of data will be written into the code sheet, and Excel can only hold 65,536 rows of data. Keeping the value below the limit can be done by changing the value in the cell ***"C26"***.

After all the inputs have been specified, clicking the ***Update*** button will instruct the program to update the data and the five plots.

In summary, the user needs to do the following steps to run the program:

1. Go to the Process Sheet Ŀ by clicking on the ***Process*** tab
2. Enter input values in the green cells
3. Adjust the value in the cell C26 so that the computed value in cell E27 is less than 65,536
4. Click the **Update** button
5. Observe the droplet profiles on the five solution plots
6. Repeat step 1 through step 5 for different input values
7. Click the **Clear Data** button to clear the data (Optional).

Additional features include the option to store the computed data into a TecPlot format file. This feature provides the user a flexibility to plot the data using other software such as TecPlot. The details about how to run this option are explained in the discussion section below.

<u>Code Sheet</u>

The last sheet is the code sheet, which contains the solution data produced by the program. Six columns namely A, B, C, D, E, and F hold the data of time, trajectory x-component, trajectory z-component, velocity x-component, velocity z-component, and drag coefficient respectively.

<u>Discussions</u>

The five plots are based on the X-Z coordinate system where X positive is to the right and Z positive is up. If the inputs of the droplet's velocity injection is set only in the X positive component, and the crossflow's velocities are set to zero, you will see a decreasing slope profile on the trajectory plot. Because relative velocity in the Z direction between the droplet and the crossflow is zero, gravity force overcomes zero buoyancy force, and results in the droplet to move down, in the negative Z direction.

The total number of cycles will determine the time for the program to process the data. ***Data taken every cycle*** also determines the speed of the program to process the data. The more data is collected, the slower the program to complete the cycle.

You are encouraged to change the chart type of the five plots which you find best and easy to analyze. You might want to switch to ***dotted points-connected line chart*** to analyze the data points.

The cells highlighted in cyan and in white are locked for the coding security purpose. It is highly recommended that initially the user not ***unlock and modify*** these cells. Any modifications made may cause the program to crash.

The last feature added to this program is the flexibility for the user to store the computed data into TecPlot format file. By enabling this option, the user needs to specify the path and the filename to store the output. This option must be ***disabled*** if the path and the file name are not specified.

Appendix A

Visual Basic Code

Under the 'Tools', 'Macro', 'Visual Basic' button, the heart of the numerical code is presented. This code is reproduced here in case something happens to the program. Its logic is similar to the FORTRAN code following, but some things are different.

```
Private Sub CommandButton3_Click()
  Dim n As Long, nn As Double, nnnn As Long, userchoice As Long
  Dim xx(0 To 1) As Double, xp(0 To 1) As Double
  Dim zz(0 To 1) As Double, zp(0 To 1) As Double
  Dim Cdm As Double, Rems As Double
  Dim A_x As Double, A_z As Double, h As Double, kl As Double, r As Double, ht As
Double
  Dim k1 As Double, k2 As Double, k3 As Double, k4 As Double, l1 As Double, l2 As
Double, l3 As Double, l4 As Double
  Dim kz1 As Double, kz2 As Double, kz3 As Double, kz4 As Double, lz1 As Double,
lz2 As Double, lz3 As Double, lz4 As Double
  Dim Urm As Double
  Dim ug As Double, wg As Double, mu As Double, rg As Double
  Dim a As Double, b As Double
  Dim location
  Dim sd, sf
  Dim a1, b1, c1, d1, e1, f1
  Dim unitconv

  Call Macro2
  Halt = False
  Msg = "Do you want to continue ?"
  Style = vbYesNo
  Title = "Jet Flow in CrossFlow"

  'Getting the input
  ug = Sheets("Process").Cells(11, 3)
  wg = Sheets("Process").Cells(12, 3)
  mu = Sheets("Process").Cells(6, 3)
  rg = Sheets("Process").Cells(9, 3)
  rd = Sheets("Process").Cells(8, 3)
  g = Sheets("Process").Cells(7, 3)
  xx(0) = Sheets("Process").Cells(17, 3)
  xp(0) = Sheets("Process").Cells(18, 3)
  zz(0) = Sheets("Process").Cells(19, 3)
  zp(0) = Sheets("Process").Cells(20, 3)
  h = Sheets("Process").Cells(10, 3)
```

```
 kl = Sheets("Process").Cells(25, 3)
 userchoice = Sheets("Process").Cells(26, 3)
 r = Sheets("Process").Cells(5, 3)
 r = r / 1000000#


   Worksheets("Process").CommandButton1.Width = 0
   Worksheets("Process").CommandButton1.Visible = True

   n = 0
   nn = -1
   nnn = 1
   ht = 0

20
   ht = ht + h
   a = xp(n)
   b = zp(n)
   Call Runge(a, b, r, rd, rg, g, mu, ug, wg, A_x, A_z)
   If (SuperExit) Then GoTo 50
   k1 = h * xp(n)
   l1 = h * A_x
   kz1 = h * zp(n)
   lz1 = h * A_z

   a = xp(n) + l1 / 2#
   b = zp(n) + lz1 / 2#
   Call Runge(a, b, r, rd, rg, g, mu, ug, wg, A_x, A_z)
   If (SuperExit) Then GoTo 50
   k2 = h * xp(n)
   l2 = h * A_x
   kz2 = h * zp(n)
   lz2 = h * A_z

   a = xp(n) + l2 / 2#
   b = zp(n) + lz2 / 2#
   Call Runge(a, b, r, rd, rg, g, mu, ug, wg, A_x, A_z)
   If (SuperExit) Then GoTo 50
   k3 = h * xp(n)
   l3 = h * A_x
   kz3 = h * zp(n)
   lz3 = h * A_z

   a = xp(n) + l3
   b = zp(n) + lz3
   Call Runge(a, b, r, rd, rg, g, mu, ug, wg, A_x, A_z)
```

```
If (SuperExit) Then GoTo 50
k4 = h * xp(n)
l4 = h * A_x
kz4 = h * zp(n)
lz4 = h * A_z

xp(n + 1) = xp(n) + (1# / 6#) * (l1 + 2# * l2 + 2# * l3 + l4)
zp(n + 1) = zp(n) + (1# / 6#) * (lz1 + 2# * lz2 + 2# * lz3 + lz4)

xx(n + 1) = xx(n) + (1# / 6#) * (k1 + 2# * k2 + 2# * k3 + k4)
zz(n + 1) = zz(n) + (1# / 6#) * (kz1 + 2# * kz2 + 2# * kz3 + kz4)

nn = nn + 1#

If nn = 0 Then
  Urm = (((xp(n) - ug) ^ 2) + ((zp(n) - wg) ^ 2)) ^ 0.5
  Rems = 2 * rg * Urm * r / mu
  If Rems <= 1000 Then
    Cdm = (((Rems ^ (2 / 3)) / 6) + 1) * 24 / Rems
  Else
    Cdm = 0.424
  End If
  Sheets("Code").Cells(2, 1) = nn * h
  Sheets("Code").Cells(2, 2) = xx(n)
  Sheets("Code").Cells(2, 3) = zz(n)
  Sheets("Code").Cells(2, 4) = xp(n)
  Sheets("Code").Cells(2, 5) = zp(n)
  Sheets("Code").Cells(2, 6) = Cdm
  Worksheets("Process").CommandButton1.Width = ((nn / kl) * 165.75)
  Worksheets("Process").CommandButton1.Caption = ((nn / kl) * 100) & "%"
  Worksheets("Process").CommandButton1.Height = 20.25

Else
  If 0 = (nn Mod userchoice) Then
    nnn = nnn + 1
    Urm = (((xp(n + 1) - ug) ^ 2) + ((zp(n + 1) - wg) ^ 2)) ^ 0.5
    Rems = 2 * rg * Urm * r / mu
    If Rems <= 1000 Then
      Cdm = (((Rems ^ (2 / 3)) / 6) + 1) * 24 / Rems
    Else
      Cdm = 0.424
    End If
    Sheets("Code").Cells(nnn + 1, 1) = nn * h
    Sheets("Code").Cells(nnn + 1, 2) = xx(n + 1)
    Sheets("Code").Cells(nnn + 1, 3) = zz(n + 1)
    Sheets("Code").Cells(nnn + 1, 4) = xp(n + 1)
```

```
        Sheets("Code").Cells(nnn + 1, 5) = zp(n + 1)
        Sheets("Code").Cells(nnn + 1, 6) = Cdm
        Worksheets("Process").CommandButton1.Width = ((nn / kl) * 165.75)
        Worksheets("Process").CommandButton1.Caption = ((nn / kl) * 100) & "%"
        Worksheets("Process").CommandButton1.Height = 20.25

        DoEvents
        If (Halt) Then
         DoEvents
         Halt = False
         Response = MsgBox(Msg, Style, Title)
          If Response = vbNo Then
           GoTo 10
          Else
          End If
        End If

      Else
      End If
    End If


    xp(n) = xp(n + 1)
    zp(n) = zp(n + 1)
    xx(n) = xx(n + 1)
    zz(n) = zz(n + 1)

    If nn >= kl Then
      GoTo 10
    Else
      GoTo 20
    End If

10
  Macro1
  sd = Sheets("Process").Cells(27, 5)
  If Optional_Save = "YES" Then
   location = Sheets("Process").Cells(44, 2)
   Open location For Output As #1
   Print #1, "TITLE = "; Spc(2); """"; "Spray Jet In CrossFlow"; """"
   If Unit = "millimeter" Then
    unitconv = 1000#
    Print #1, "Variables ="; Spc(2); """"; "Time(sec)"; """"; Spc(2); """"; "X(mm)"; """";
Spc(2); """"; "Z(mm)"; """"; Spc(2); """"; "U(mm/s)"; """"; Spc(2); """"; "W(mm/s)";
""""; Spc(2); """"; "Cd"; """"
   Else
```

```
   unitconv = 1#
   Print #1, "Variables ="; Spc(2); """"; "Time(sec)"; """"; Spc(2); """"; "X(meter)"; """";
Spc(2); """"; "Z(meter)"; """"; Spc(2); """"; "U(m/s)"; """"; Spc(2); """"; "W(m/s)"; """";
Spc(2); """"; "Cd"; """"
  End If
  Print #1, "ZONE I ="; sd & ","; Spc(2); "F = POINT"
  For sf = 2 To sd + 1
   a1 = Sheets("Code").Cells(sf, 1)
   b1 = Sheets("Code").Cells(sf, 2)
   b1 = b1 * unitconv
   c1 = Sheets("Code").Cells(sf, 3)
   c1 = c1 * unitconv
   d1 = Sheets("Code").Cells(sf, 4)
   d1 = d1 * unitconv
   e1 = Sheets("Code").Cells(sf, 5)
   e1 = e1 * unitconv
   f1 = Sheets("Code").Cells(sf, 6)
   Print #1, a1; Spc(2); b1; Spc(2); c1; Spc(2); d1; Spc(2); e1; Spc(2); f1
  Next sf
  Close #1
 Else
 End If
50
SuperExit = False
End Sub


Sub Runge(xps, zps, rs, rds, rgs, g, mus, ugs, wgs, A_xs, A_zs)
 Dim Area As Double, Volume As Double, Ur As Double, Re As Double, Cd As Double

 Area = (22# / 7#) * rs ^ 2
 Volume = (22# / 7#) * (4# / 3#) * rs ^ 3

 Ur = (((xps - ugs) ^ 2#) + ((zps - wgs) ^ 2#)) ^ 0.5
 Msg = "The Time Step Size is too big. Program Terminated Ur = " & Ur
 Msg2 = "Reduce the time step size!"
 Style = vbOKOnly
 Title = "Jet Flow in CrossFlow"

 If Ur > (10 ^ 80) Then
  Response = MsgBox(Msg, Style, Title)
  Response = MsgBox(Msg2, Style, Title)
  SuperExit = True
  Sheets("Process").Range("C10").Select
 Else: End If
 Re = 2# * rgs * Ur * rs / mus
```

```
    If Re <= 1000 Then
      Cd = Re ^ (2# / 3#)
      Cd = Cd / 6#
      Cd = (Cd + 1#) * 24# / Re
    Else
      Cd = 0.424
    End If
    A_xs = -0.5 * rgs * Ur * Area * Cd * (xps - ugs) / (rds * Volume)
    A_zs = (-0.5 * rgs * Ur * Area * Cd * (zps - wgs) + (rgs - rds) * Volume * g) / (rds *
Volume)
End Sub
```

Appendix B

Equivalent Fortran Program

```
*deck main
c==================================================================*
c                              *
c This program calculates droplet's trajectory and its velocity *
c after it is injected into a cross-stream using the 4th-order  *
c Runge-Kutta method. It calls derivs which defines the ordinary *
c differential equations (ODEs).                  *
c All the constant values can be found in subroutine derivs  *
c==================================================================*
c
   parameter (nd=10000000)
   dimension x(nd),dx(nd),u(nd),du(nd),t(nd),xt(nd),ut(nd),
   &     dxt(nd),dut(nd),dxm(nd),dum(nd)
   dimension z(nd),dz(nd),w(nd),dw(nd),zt(nd),wt(nd),
   &     dzt(nd),dwt(nd),dzm(nd),dwm(nd)
c +++
c +++ input parameters
c +++
   print*,'Enter time step :'
   read*,dt
   print*,' '
   print*,'Enter total time:'
   read*,tend
   print*,'Enter initial droplet position x(0) :'
   read*,x(1)
   print*,' '
   print*,'Enter initial droplet velocity u(0) :'
   read*,u(1)
   print*,' '
   print*,'Enter initial droplet position z(0) :'
   read*,z(1)
   print*,' '
   print*,'Enter initial droplet velocity w(0) :'
   read*,w(1)
   print*,' '
   print*,'Enter radius [micro-meter]:'
   read*,r
   print*,' '
   r=r*1.e-6
c
   pi=acos(-1.)
   ad=pi*r**2      ! Droplet area
   vd=4./3.*pi*r**3    ! Droplet volumn
c
   print*,'Area  = ',ad
   print*,'Volumn = ',vd
c
   t(1)=0.0
c
   n=tend/dt
c
```

```
      write(*,*)
      write(*,*)' time step dt = ',dt
      write(*,*)

      if(n.gt.nd)then
       write(*,*)' n is greater than nd -- run aborting'
       write(*,*)' n = ',n,' nd = ',nd
       stop
      endif
c
      t(n)=tend
c +++
c +++ Solving the differential equations using 4th-order
c +++ Runge Kutta methods.
c +++
      dtt=0.5*dt
      do 10 i=1,n-1
c +++
c +++
       call derivs(i,r,ad,vd,u,dx,du,w,dz,dw)
c +++
c +++
       xt(i)=x(i)+dtt*dx(i)
       ut(i)=u(i)+dtt*du(i)
       zt(i)=z(i)+dtt*dz(i)
       wt(i)=w(i)+dtt*dw(i)
       call derivs(i,r,ad,vd,ut,dxt,dut,wt,dzt,dwt)
c +++
c +++
       xt(i)=x(i)+dtt*dxt(i)
       ut(i)=u(i)+dtt*dut(i)
       zt(i)=z(i)+dtt*dzt(i)
       wt(i)=w(i)+dtt*dwt(i)
       call derivs(i,r,ad,vd,ut,dxm,dum,wt,dzm,dwm)
c +++
c +++
       xt(i)=x(i)+dt*dxm(i)
       ut(i)=u(i)+dt*dum(i)
       dxm(i)=dxt(i)+dxm(i)
       dum(i)=dut(i)+dum(i)
       zt(i)=z(i)+dt*dzm(i)
       wt(i)=w(i)+dt*dwm(i)
       dzm(i)=dzt(i)+dzm(i)
       dwm(i)=dwt(i)+dwm(i)
       call derivs(i,r,ad,vd,ut,dxt,dut,wt,dzt,dwt)
c +++
c +++ advance the solutions to n+1 time step
c +++
       t(i+1)=t(i)+dt
       x(i+1)=x(i)+dt/6.*(dx(i)+dxt(i)+2.*dxm(i))
       u(i+1)=u(i)+dt/6.*(du(i)+dut(i)+2.*dum(i))
       z(i+1)=z(i)+dt/6.*(dz(i)+dzt(i)+2.*dzm(i))
       w(i+1)=w(i)+dt/6.*(dw(i)+dwt(i)+2.*dwm(i))
c
   10 continue
c
c +++
```

```
c +++ write data sets to file result.dat
c +++
   open(unit=3,file='result.dat',form='formatted')
   do 20 i=1,n
    write(3,100) t(i),x(i),z(i),u(i),w(i)
  20 continue
c
   write(*,*)' The data is written to result.dat as'
   write(*,*)' t, x, z, u, w'
c
   close(3)
c
 100 format(1x,f5.3,4(1x,f15.10))
   end
c
c===========================================================
*deck derivs
   subroutine derivs(i,r,ad,vd,u,dx,du,w,dz,dw)
c -------------------------------------------------------*
c This subroutine defines the differential equations   *
c being solved by the program.              *
c                             *
c -------------------------------------------------------*
   dimension u(1),dx(1),du(1),w(1),dz(1),dw(1)
c +++
c +++ Constant values
c +++
   rhod=300.    ! Droplet density
   rhog=1.19    ! Fluid density
   ug=0.      ! Fluid velocity in x-direction
   g=9.8      ! Gravity
   wg=-5.0     ! Fluid velocity in z-direction
   eni=0.0004    ! viscosity of the gaseous fluid
c
   ur=abs(sqrt((u(i)-ug)**2+(w(i)-wg)**2))
c
   red=2.*r*ur*rhog/eni
   if (red.le.1000.) then
    cd=24./red*(1.+1./6.*red**(2./3.))
   else
    cd=0.424
   endif
c
   dx(i)=u(i)
   du(i)=1./(rhod*vd)*(-0.5*rhog*(u(i)-ug)*ur*ad*cd)
c
   dz(i)=w(i)
   dw(i)=1./(rhod*vd)*(-0.5*rhog*(w(i)-wg)*ur*ad*cd+
   &   (rhog-rhod)*vd*g)
c
   return
   end
```

References

Leong, M.Y., McDonell, V.G., and Samuelsen, G.S., 2000, “Mixing of an Airblast-Atomized Fuel Spray Injected Into a Crossflow of Air,” NASA/CR—2000-210467.

Leong, M.Y., McDonell, V.G., and Samuelsen, G.S., 2001, “Effect of Ambient Pressure on an Airblast Spray Injected into a Crossflow,” J. Prop. Power, Vol. 17, No. 5, Sept.–Oct., pp. 1076–1084.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 2002	Technical Memorandum

4. TITLE AND SUBTITLE

An Interactive Excel Program for Tracking a Single Droplet in Crossflow Computation

5. FUNDING NUMBERS

WU–708–87–13–00

6. AUTHOR(S)

E. Urip, S.L. Yang, and C.J. Marek

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field
Cleveland, Ohio 44135–3191

8. PERFORMING ORGANIZATION REPORT NUMBER

E–13459

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM—2002-211710

11. SUPPLEMENTARY NOTES

E. Urip and S.L. Yang, Michigan Technological University, Mechanical Engineering, Engineering Mechanics Department, 1400 Townsend Drive, Houghton, Michigan 49931–1295; C.J. Marek, NASA Glenn Research Center. Along with this report, there is an interactive Excel spreadsheet that is not published on the web. To obtain a copy of the spreadsheet contact C.J. Marek. Responsible person, C.J. Marek, organization code 5830, 216–433–3584.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category: 07 Distribution: Nonstandard

Available electronically at http://gltrs.grc.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 301–621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Spray jet in crossflow has been a subject of research because of its wide application in systems involving pollutant dispersion, jet mixing in the dilution zone of combustors, and fuel injection strategies. The focus of this work is to investigate dispersion of a 2-dimensional atomized spray jet into a 2-dimensional crossflow. A quick computational method is developed using available software. The spreadsheet can be used for any 2D droplet trajectory problem where the drop is injected into the free stream eventually coming to the free stream conditions. During the transverse injection of a spray into high velocity airflow, the droplets (carried along and deflected by a gaseous stream of co-flowing air) are subjected to forces that affect their motion in the flow field. Based on the Newton's Second Law of motion, four ordinary differential equations were used. These equations were then solved by a fourth-order Runge-Kutta method using Excel software. Visual basic programming and Excel macrocode to produce the data facilitate Excel software to plot graphs describing the droplet's motion in the flow field. This program computes and plots the data sequentially without forcing users to open other types of plotting programs. A user's manual on how to use the program is also included in this report.

14. SUBJECT TERMS

Droplet; Trajectory; Excel spreadsheet

15. NUMBER OF PAGES

29

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	